ECES 338 Assignment #1 Due: February 5, 1999

Spring 99 (100 points)

Ozsoyoglu,G.

In this problem you will spawn and execute processes concurrently under the Unix operating system. The basic Unix system calls which you will be required to use are *fork(), execl(), getpid(), getppid(), getlogin(), getrusage(), exit() and wait().* All the Unix system calls that you need are explained in detail in the Unix book by Gray, chapters 2 and 3. Also, you can use the man pages of the Unix OS.

The purpose of this assignment is to create three concurrently executing processes, and have them execute simple Unix system calls. You(r program when executed, i.e., the "parent process") will create two Unix child processes by using the fork() Unix system call. You will then synchronize the execution of these two child processes by using the sleep() system call.

The parent process will initialize two variables f1 and f2 (for fibonacci numbers 1 and 2) to 1 and 1, respectively, and then fork two child processes, and wait for a long enough time (sleep() call) for the child processes to compute fibonacci numbers by alternation. More specifically, child 1 and child 2 will compute and print the "next" fibonacci number alternatively using the recursive formula

$$f_i = f_{i-1} + f_{i-2}.$$

1) Find out at execution time, your parent and child processes' real and effective user ids, and their group ids. **Question:** Explain briefly the purpose of these ids.

2) Have the parent process find out your login name using getlogin(), and print it.

3) For both the parent process and the child processes, at the end of their execution, find and print the wall clock time, user CPU time, and system CPU time that they use. (Have your processes loop for a while in order to accumulate some nonzero time values.) The unix function getrusage( ) can be used for this purpose. Note that you must include Berkeley libraries (ucb) when compiling your program.

4) Force the parent process to wait for the termination of child processes, and have the parent process print the termination status of the child process.

5) Finally, have the parent process perform execl(), and execute the binary "df".(**Question**: What is the df command?) The system call execl() requires the absolute path name of the executable file df, which is "/bin". Also the last parameter to the execl should be a 0.

Use the Unix command script to record the entire session (try "man script"), and submit the typescript file as well as your code. Please remember to produce meaningful print statements like "The process id xxx produced the fibonacci number f3 as yyy", etc., instead of just numbers. Also, please turn in separately your answers to the questions above.